

Exchange Document Development Process: An Overview



Scott Came
President and Principal Consultant
Justice Integration Solutions, Inc.

December 30, 2004

The purpose of this document is to propose a basic process that can be followed when creating Global Justice XML Data Model (GJXDM)-conformant Exchange Documents* to govern the exchange of data between justice partners at any jurisdictional level.

The focus of this process is on the technical/mechanical steps used to proceed from requirements to a schema solution; this document does not address the important issues of governance associated with the vetting and approving of reference document content.

The process described in this document is based on the process and methodology that were used to develop the first draft of a sentence order reference document, starting with a workshop held at the Colorado AOC in June, 2004, and continued by a small workgroup following the workshop. The process has been further refined with experiences from subsequent workshops, including the Law Enforcement GJXDM Reference Document Workshop held at the Los Angeles County Information Systems Advisory Body November 15-17, 2004. It has also benefited significantly from feedback provided by the members of the IJIS Institute XML Advisory Committee.

This document assumes a basic understanding of XML, familiarity with the concept of XML schemas, and a basic understanding of GJXDM. Please see the references section at the end for resources on these topics.

The remainder of this document will detail a process that involves the following basic steps:

1. Assemble a workgroup to develop the exchange document; this group should include business experts as well as technical members familiar with domain object modeling and GJXDM
2. Build a domain model (a graphical depiction of the document structure) from existing paper forms as well as business experts' knowledge, leveraging GJXDM where appropriate without being constrained by GJXDM
3. Map the entities and properties in the domain model to GJXDM, noting extensions where they occur
4. Build a GJXDM schema package from the mapping, using available tools (like the online subset schema generator and stylesheets) to create subset, constraint, extension, and document schemas
5. Develop sample instances, document business rules, and package the exchange document artifacts

* A clarification of terminology is in order here. This paper will leverage definitions of the terms *Exchange Document* and *Reference Exchange Document* as provided by the Global XML Structure Task Force (XSTF). These terms are defined as follows. An Exchange Document is a collection of artifacts that specify the structure and content of a GJXDM-conformant XML document instance for a particular justice information exchange. A Reference Exchange Document is an Exchange Document that has been mandated, approved, endorsed, recommended, or acknowledged by a cognizant organization in the justice community, and may be used as the basis for an Exchange Document meeting the specific business needs of an information sharing enterprise. While this paper was initially conceived as a proposed process for developing Reference Exchange Documents, it is equally applicable and appropriate for the development of Exchange Documents in any jurisdiction. Consequently, this paper will use the term *Exchange Document* throughout.

Deliverables and Milestones

The following deliverables result from this process, according to the following approximate timeline. This timeline is provided to *suggest* the *approximate* amount of time required to complete an exchange document; however, each document will be different and so the schedule for developing each will be different. Note that these estimates are based on experience with sentence orders, which are more complex than many exchange documents. Consequently, this schedule can be viewed as conservative for less complex documents.

Deliverable	Team-hours to reach baseline	Participating roles
Domain model—face-to-face workshop	16	<ul style="list-style-type: none">• Business experts• Domain modeler• GJXDM experts
Domain model—refinement	32	<ul style="list-style-type: none">• Business experts• Domain modeler• GJXDM experts
GJXDM Mapping	40	<ul style="list-style-type: none">• GJXDM experts
Schema package	16	<ul style="list-style-type: none">• GJXDM experts
Examples	8	<ul style="list-style-type: none">• Business experts• GJXDM experts

Note that we use the concept of “baselining” deliverables here, rather than “completing” them. On an exchange document development project, artifacts continue to be revised over time, and so are never really “complete.” However, the intent of this table is to identify approximate times to reach a stable baseline of each artifact, where major changes were no longer being made. “Team hours” means elapsed time devoted to the deliverable work by the identified participating roles.

Each deliverable (and the process used to create it) will be described in subsequent sections.

Workgroup Composition

The experience of the sentence order workgroup indicates that having the right participation from the right kinds of experts is key to the success of an exchange document development project. In particular, the following areas of expertise need to be on the development team:

- A representative group of subject-matter and business experts, who fully understand the needs of the justice partners that will be exchanging the data represented by the exchange document; in addition to sharing their expertise, these experts could also supply examples of paper documents currently exchanged between the partners

- A person or persons skilled in domain (object) modeling, who can facilitate the process of documenting the data structures in terms of business objects and their relationships; this role requires adequate experience with modeling methodologies and tools
- A person or persons who understand GJXDM well enough to map the business objects to structures in GJXDM

In the next section, we will describe what we mean by “domain modeling” and “business objects,” and will also offer some thoughts on tools appropriate for this task.

It is important to note that the critical roles identified above need not be supplied by different people. In particular, the domain modeling role and GJXDM mapping role can often be played by the same person.

The domain modeler / facilitator should encourage the participating business experts to practice good listening techniques in the modeling sessions, and not to limit their thinking to their own data or business processes.

Domain Model

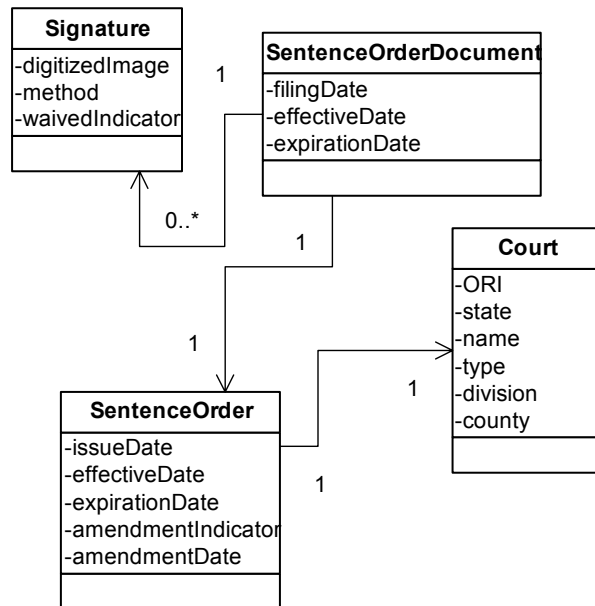
A “domain model” is a graphical depiction of the data structures involved in the exchange document and the relationships between them. It is quite similar to entity-relationship modeling common in database analysis/development, but adds additional mechanisms for identifying object-oriented relationships (like inheritance) that are important when the end-product is an XML schema. The structures depicted on the domain model diagram are classes, but we think of them in particular scenarios, so we call them “business objects.” Consequently, sometimes we call the model an “object model” or “domain object model”...these are equivalent to the term “domain model.”

Experience with complex domain models has demonstrated the importance of a high-level graphical picture of the model, to supplement the detailed model described below. This diagram (or small set of diagrams) describes the most significant components in the exchange document. In many cases, these components will correspond to major sections or topical areas in the paper documents, or will encapsulate logical groupings of entities in the detailed domain model. This high-level picture can be created in UML, but need not be in any formal notation (some workgroups have had success with Visio or PowerPoint slides.)

The domain model artifact itself (i.e., the diagram) can be as simple or complex as the situation requires. Some exchange document workgroups have had some success with very simple model diagrams, drawn on PowerPoint slides with simple circles and arrows. This technique should work fine for relatively simple exchange documents with few entities. For more complex documents (like sentence orders), with many entities and complex relationships (including inheritance relationships), using a Unified Modeling Language (UML) class diagram to create the domain model is probably more appropriate. UML is an open industry standard that is

familiar to many software developers, analysts, and designers. It has well-defined structures for describing the kinds of class and object relationships that occur in exchange documents, so the workgroup need not invent notations. UML is supported by a wide range of tools, including Microsoft Visio, and including several that are available under a no-cost or open-source license.

As an example, the following picture is a snippet of the sentence order domain model in a UML class diagram:



In addition to the high-level and detailed (UML) domain model diagrams, a text-based listing of domain model elements (and their definitions) can be an effective communication tool, especially for stakeholders who find UML diagrams less intuitive. Some workgroups have had success representing the element list in a spreadsheet; if organized well, this can be the same spreadsheet used for mapping (described later in this document.) It is recommended that the model elements appear in the listing in the same order they appear in the document hierarchy (or as close to it as possible.) Additional standardization is needed in this area to establish conventional notations for representing object-oriented concepts like inheritance.

It is important for the domain modeler to produce a candidate domain model prior to the start of the workgroup's first session. If the domain modeler lacks adequate subject-matter expertise, one of the workgroup's business experts should be available to assist in the construction of this initial model.

Note that in at least a few cases, UML diagrams (or at least partial ones) for exchange document domains may already be available from previous exchange projects. Reusing these existing models can significantly accelerate the development process, although it is important to review the models within the workgroup to assess their correctness and appropriateness for the exchange document under development.

The domain modeling effort is much more successful if the workgroup can participate in building the model in a face-to-face setting, at least for the bulk of the effort. For a exchange document of significant complexity, a two-day workshop (like the sessions held in Denver in June, 2004) is typically adequate. Once the model is fairly stable (with the most significant structures defined), remote modeling (emails and conference calls) facilitated by the domain modeler/analyst can work fairly well.

The domain modeler should prepare the other workgroup members to comprehend the modeling notation, prior to the start of the first modeling session (or at least at the very beginning of the session.) If the chosen modeling notation is UML, then a brief introduction to UML will help stakeholders who are unfamiliar with the technology to be more effective participants in the sessions.

If multiple technical facilitators or domain modelers are participating in the workgroup, select a modeling notation with which they are all comfortable. Also ensure that all facilitators and modelers are comfortable with the candidate domain model developed prior to the first session, and that they are comfortable with the choice of modeling tool.

One of the important lessons learned from document development experiences has been that taking the time and effort to model the document business domain is very helpful. This step provides the following benefits:

- The model helps to focus the workgroup participants on a single artifact that defines terminology and the relationships between data elements—both of which are important in creating a GJXDM-conformant schema later
- The model provides a fairly simple, graphical depiction of the data structures and their relationships, which is easier than a schema for most business experts to use in defining the initial document structure; use of a simple model can accelerate the communication and development process within the workgroup
- The model is technology-neutral, and so can be leveraged in other areas of data exchange development, once the schema development is complete

GJXDM Mapping

Once the model is defined, whether using circles on a PowerPoint slide or a UML class diagram, the next step is to map the structures in the model to the GJXDM. This mapping is accomplished in a simple Excel spreadsheet with three columns. The first two columns indicate the class and property (from the model diagram) that contain the data to be mapped; the third column contains a “path” through the GJXDM that points to the GJXDM element that corresponds to the property. Various

strategies can be used for defining this path; in the sentence order workgroup, it worked best to use element names at each point in the hierarchy, and to separate levels with slashes (“/”).

Additional columns could be added to the spreadsheet to contain notes, such as hints about inherited properties in the path, and also whether the workgroup would like to suggest modifications to the GJXDM to make the mapping fit better.

If it is not possible to map a class/property from the domain model to something in GJXDM, then the spreadsheet should indicate that that class/property requires an extension to GJXDM. In the sentence order workgroup, such properties were identified by shading the path yellow to distinguish them easily.

The following is an example snippet from the sentence order mapping spreadsheet:

Class	Property or relationship	GJXDM Path
SentenceOrder Document	filingDate	Document/DocumentDescriptiveMetadata/DocumentFiledDate
	effectiveDate	Document/DocumentDescriptiveMetadata/DocumentEffectiveDate
	expirationDate	Document/DocumentDescriptiveMetadata/DocumentExpirationDate
	SentenceOrder	SentenceOrderDocument/SentenceOrder
	Signature	SentenceOrderDocument/Signature

It is worth noting that many UML modeling tools support “metadata” for each class and property in a class diagram. This should make it possible to perform the GJXDM mapping entirely within the UML modeling tool (if the workgroup is using one), and perhaps even to generate an initial “want list” file that can be imported into the online tool for subset schema generation (see subsequent section for the details of subset schema generation.) The opportunities for using UML tool metadata for performing GJXDM mapping is certainly worthy of further investigation by the justice community. If proved viable, this approach would remove the need for creating a mapping spreadsheet, which would make it a bit easier to keep the model and mapping synchronized.

Aside: Using GJXDM Types in the Domain Model

In the previous two sections, we described a two-step process, in which a domain model is created first, followed by a mapping of that model to the GJXDM. This process raises an important question: to what extent should the workgroup use GJXDM types in the domain model?

The simple answer to this question is: as much as practical consistent with meeting the specific business needs for data elements and data structure.

It is very important for the domain model to have integrity. The model is a communication tool used by the business experts to define the structure of the data to be exchanged. For it to be an effective communication tool, it is very important that it contain structures that fit very well with the

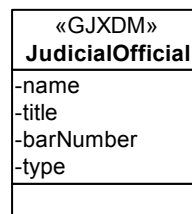
business experts' shared understanding of the problem. In many cases, GJXDM structures will be a good fit; in others, they may not.

The GJXDM is certainly a very good source of potential domain objects, and utilizing GJXDM components "out of the box" can assist in providing a baseline for commonality across jurisdictions or domains. The workgroup would be wise to search in GJXDM for well-fitting objects first, but should not feel constrained to use GJXDM types. In cases where a GJXDM structure seems like a good fit, it is important for the workgroup to review the GJXDM type's definition to make sure. (One of the GJXDM best practices is to use a type from GJXDM *only* when the definition fits the information being exchanged.)

It is also important to be pragmatic in accomplishing the goals of the workgroup. The end product of this process is a GJXDM-conformant schema that effectively accomplishes the objectives of the data exchange. If the domain model diverges unnecessarily from GJXDM (due, for example, to lack of understanding of GJXDM), then the crucial mapping exercise will be unnecessarily difficult. Therefore, if a structure in GJXDM fits the domain well, then the workgroup is much better off using that structure in the domain model rather than inventing something new. (This is why it is very helpful to have business expertise, domain modeling expertise, and GJXDM expertise in the same room during the modeling sessions, to leverage the structures already in GJXDM whenever possible.)

Finally, as the first several standard exchange documents are developed, the expectation is that the justice community will learn important things about GJXDM, including where GJXDM needs to be improved. Building a domain model that is unrestrained by GJXDM will help identify areas for improvement, and conversely will help to confirm where GJXDM already fits the domain well.

Where GJXDM types are used in the domain model, and where the workgroup is using UML to depict the model, it would be a good idea to use a UML stereotype to identify the type as such. (The sentence order workgroup did not do this, but it would be an improvement for future workgroups to do so.) A stereotype appears at the top of a class, surrounded by guillemots, like this:



After the GJXDM mapping is complete, the next step is to create the “schema package.” A exchange document schema is actually several schemas that are combined to define the structure of XML instance documents. These schema types (and the process used to create them) are described in this section.

GJXDM Subset Schema

The GJXDM is actually a *reference schema*, and need not be used in its entirety to define instance document structure. Because of the size and complexity of the full GJXDM schema, it is generally a good idea, when creating a schema for a particular exchange, to create a “subset schema” that removes unused GJXDM components.

Subset schemas must be crafted carefully, and subsets must adhere to a few simple but important rules. (These rules are defined online...see the references at the end of the paper for a pointer.) To make sure that a proper subset is created, it is a good idea to use the subset schema generation tool developed by Georgia Tech Research Institute and funded by the Justice Department’s Office of Justice Programs. This was the approach taken by the sentence order workgroup. A link to this online tool is provided in the references section at the end of this document.

Using the paths in the mapping spreadsheet described above, each required type and element in GJXDM is selected in the subset generation tool. It is important to include types that will be extended as well (as indicated by the yellow-shaded cells in the spreadsheet.) Once all the needed types and elements are selected, the tool is used to generate a subset schema and any dependant schemas. Fortunately, the tool saves a “want list” every time it generates a schema, so it is possible to add types and elements later by uploading the wantlist document into the tool. Workgroups should expect the subset generation process to require a few iterations, as it is difficult to get the subset correct on the first try; the wantlist document makes this kind of iterative process possible.

Constraint Schema

The subset schema generated by the online tool contains the types, elements, and basic relationships needed for the exchange document. However, the subset generation tool does not currently provide a way to specify the *cardinality* for relationships (e.g., one-to-one, one-to-many). Since the cardinality of all relationships in GJXDM is zero-to-many (basically, “anything goes”), a typical exchange document will require a tightening down of the GJXDM defaults. For example, exchange documents will typically need to require certain data elements, or limit data elements to one occurrence (rather than many).

Implementing these requirements necessitates a modification of the subset schema generated by the online tool. The nature of the modification is to alter the element definitions’ cardinality attributes (`minOccurs` and `maxOccurs`) as appropriate. Because the subset may change (for example, as new requirements are uncovered and the subset is re-generated), it is desirable to define the cardinality modifications in

such a way as to make the changes repeatable and not subject to human error. That is, it is desirable not to reapply the changes manually and from memory each time the subset changes.

The sentence order workgroup solved this problem by developing an XSL stylesheet that transforms the generated subset schema into a “constraint schema” that has the “tightened down” cardinality constraints. This stylesheet is edited once and saved, and can be reapplied to future generated subsets as needed. The initial version of the stylesheet is generated from the subset by yet another stylesheet; this second stylesheet is a general solution to the problem and can be used for any exchange document, not just sentence orders.

Extension Schema

The constraint schema contains types and elements from GJXDM only. A second schema is required to define those types and elements that the mapping identified as extensions. This second schema is called an extension schema. It is created from scratch by hand. It imports the constraint schema and any other schemas required by the extension types and elements. The extension schema defines a new namespace, the name of which should indicate in some way the domain of the exchange document.

Document Schema

The document schema is very short and simple. It indicates which element from the extension schema is the root element of the exchange document instance. The document schema defines its own namespace, and imports the extension schema.

Example Exchange Document Usages

To complete the exchange document deliverables, a workgroup should create at least one example instance document. This will help the team test the correctness of the schema package (e.g., to make sure the schemas all have the correct relationships.) It will also provide an illustration of what a concrete instance looks like, rather than the more abstract structure represented in the schemas.

Depending on the project and workgroup, it also might be helpful to build a Web Services Description Language (WSDL) document to define SOAP web services messages to represent the exchange between partners. Some workgroups have also found a stylesheet useful, to transform instances into a “publishable” or user-friendly format like HTML or PDF.

Conclusion

In this paper, we have described a simple, three-step process for building exchange document schema packages, starting with business expertise and paper samples:

1. Organize a team and build a domain model from sample paper exchange documents and business experts' knowledge (utilizing GJXDM types but not being constrained by them in describing the structure of the information)
2. Map the entities (classes) and properties in the domain model to the GJXDM, noting extensions where necessary
3. Use the online GJXDM subset schema generator tool to build a proper GJXDM subset schema, and build constraint, extension, and document schemas manually (possibly aided by stylesheets to help with subset schema transformations)

The document schema (and related schemas) output from this process define valid structure for XML instance documents. They can also be used directly in WSDL documents that define the rules for standard web services (SOAP) exchanges of the documents.

References

The following references may prove useful during planning and execution of a exchange document development process.

If your workgroup decides on the UML approach to documenting the domain model, Martin Fowler's *UML Distilled* (Addison-Wesley, 2003) is a nice introduction, and includes two short chapters on class diagrams. For a more in-depth discussion of domain modeling techniques, Eric Evans' *Domain-Driven Design* (Addison-Wesley, 2003) provides thorough coverage of the concepts and techniques.

When creating the schema package, it is helpful to have a good XML Schema reference handy. Eric van der Vlist's *XML Schema* (O'Reilly and Associates, 2002) is a good one.

David Carlson's *Modeling XML Applications with UML* (Addison-Wesley, 2001) is a more extensive exploration of the idea of modeling XML vocabularies with UML, the basics of which are embodied in the process described in this paper. Carlson's website is also a good reference in this area (<http://www.xmlmodeling.com>).

Resources for GJXDM can be found online at http://it.ojp.gov/topic.jsp?topic_id=43 and <http://justicexml.gtri.gatech.edu/index.html>. The subset schema generation tool is at http://justicexml.gtri.gatech.edu/subset_tool.html. GJXDM itself is at <http://it.ojp.gov/jxdm/>. When performing the mapping of the domain model to GJXDM, it is very helpful to have a tool that can search the GJXDM from different "angles"; the Wayfarer tool on the National Center for State Courts website (http://www.ncsconline.org/d_tech/gjxdm/) does this well. There is also a model search/viewer tool built into the GTRI subset schema generator.